---------- CEN 6070 Software Testing & Verification ----------

Exam 1 – Summer 2016 – Solution Notes

1. a. Glenford Myers ("The Psychology and Economics of Program Testing")
   b. …***the most important single thing that one can learn about testing***…*everything else that can be discussed about program testing is merely supportative in nature and icing on the cake…this vital consideration, one that seems almost trivial in nature, is the definition to the word "testing."*
   c. Testing is the process of executing a program with the *intent* of finding errors. It is based on the assumption that, "…human beings tend to be highly goal oriented, (so) establishing the *proper* goal has an important psychological effect…"
   d. **Validation testing** sometimes refers to testing undertaken to *demonstrate that a system performs correctly*. **Defect testing** sometimes refers to testing undertaken to *expose system defects.*

2. (Whereas "Test-driven" simply means that a program's design is *influenced* by testing,) "Test-first" would further imply that test cases are *written and run before* features are implemented.

3. a. Minimum Viable Quality (MVQ). It builds on the premise that some companies test their web-based software services too much before releasing them to production.
   b. (i) *Feature flighting*, also known as feature toggling or feature switching, is a technique in which a feature or set of feature-related changes can be *turned on or off (remotely)* without dependency on other code.
   (ii) Flighting directly supports MVQ by allowing for both evaluating the quality of new features and experimentation (e.g., A/B testing or comparing vCurrent to vNext). Specifically, it – together with user segmentation and telemetry – allows a quick path back to the Last Known Good Configuration (LKGC).

4. (User-) journey tests are high-level (UI) tests designed to simulate a typical user's "journey" through a complete (fully integrated) system. Such a test will typically cover a user's entire interaction with the system in achieving some end user goal. They effectively cover one path in a (multi-path) use case.

5. a. alpha testing: actual end-user testing performed **within the development environment**
   b. integration testing: testing which takes place as sub-elements are combined (i.e., *integrated*) to form higher-level elements
   c. regression testing: **re-**testing to detect problems caused by the adverse effects of program change

6. c

7. (a) Black-box **boundary value analysis** (b) **pow**(x,y) function (c) The curve represents (x,y) pairs such that the actual value of x**y is equal to the **largest positive single precision floating-point number** for the given environment

(OR **overflow results if x**y exceeds +HUGE-VAL**)
The three points represent specific (x,y) values such that x**y is equal to +HUGE_VAL (on the curve), slightly larger than +HUGE_VAL (above the curve), and slightly smaller than +HUGE_VAL (below the curve)

8. a. 24, 3
   b. 6
   c. 8

9. AFCCV: (2), (3), (4), (5)
   AE:
   AFCEV: (2)
   Strategy #3: (5), (2)
   Strategy #3 plus culling rules: (2)

10. Consider a test case with inputs: A: true, B: false. This case results in the true branch being taken. Now consider another test case with inputs: A: false, B: false. This case results in the false branch being taken, but condition B never evaluates to true. Therefore, the given test cases result in branch coverage but not condition coverage, and we conclude, therefore, that branch coverage ≠> condition coverage, as desired.

11. a. When encountering a 3^{rd}-degree OR-node that must be True, the rule allows us to ignore ("cull") all combinations of input values such that more than one input is True. Thus, we consider only those 3 combinations of input condition for which exactly one input value is True. This reduces the number of combinations to consider from 7 to 3.
    b. The number of combinations to explore with an nth-degree OR-node decreases from $O(2^n)$ to O(n), more specifically, from 2^n-1 to n. The rationale is that the most important cases to explore are probably those associated with the **minimally sufficient conditions** (exactly one incoming True edge) for the OR node to evaluate to True.

12. The problem is with measures such as "number of errors" or "number of bugs" being ill-defined and therefore difficult to interpret objectively. The program, which had no body, makes it clear that such measures can be difficult to interpret.

13. a. performing reviews on critical documents like requirements
    b. capture/playback tools and structural coverage tools

14.

variable	du-pair	du-path(s)
A	(1,2)	<1,2>
	(1,3)	<1,3>
	(1,<1,2>)	<1,2>
	(1,<1,3>)	<1,3>
	(2,2)	<2,2>
	(2,3)	<2,3>
	(2,<2,2>)	<2,2>
	(2,<2,3>)	<2,3>
B	(1,2)	<1,2>, <1,2,2>
	(1,3)	<1,2,3>, <1,2,2,3>, <1,3>

15. a. $X_0 \leq 0$ & $Y_0 - X_0 > 0$ & $X_0 + 1 > 0$ & $Y_0 - X_0 - (X_0 + 1) > 0$ & $X_0 + 1 > 0$ & $Y_0 - X_0 - 2(X_0 + 1) \leq 0$
    b. (0,2)

16. b

17. F, G, A, K, B, C, H

18. e

19. a. is; b. is not; c. is; d. is; e. is not; f. is

20. f

Histogram of Raw Scores